Vietnam Journal of Mathematics
https://doi.org/10.1007/s10013-018-0311-1

Recent Advances in Quadratic Programming Algorithms for Nonlinear Model Predictive Control

Dimitris Kouzoupis[1] · Gianluca Frison[1] · Andrea Zanelli[1] · Moritz Diehl[1]

Received: 19 December 2017 / Accepted: 11 July 2018 / Published online: 29 September 2018

Abstract
Over the past decades, the advantages of optimization-based control techniques over conventional controllers inspired developments that enabled the use of model predictive control (MPC) in applications with very high sampling rates. Since at the heart of most linear and nonlinear MPC controllers resides a quadratic programming (QP) solver, the implementation of efficient algorithms that exploit the underlying problem structure drew the attention of many researchers and the progress in the field has been remarkable. The aim of this paper is to summarize the main algorithmic advances in the field and to provide a consistent benchmark between a selection of software tools that have been recently developed. The code that was used for the simulations is publicly available for readers that wish to reproduce the results or test the benchmarked solvers on their own nonlinear MPC applications.

Keywords Quadratic programming · Optimal control · Numerical methods

Mathematics Subject Classification (2010) 90C20 · 49J20 · 49M15

1 Introduction

Model predictive control (MPC) is an advanced control strategy that can naturally handle constrained systems with multiple inputs and outputs [57, 69]. However, the computational cost associated with an MPC controller is typically much higher than in conventional control techniques, since a new optimization problem needs to be solved at each sampling time. This limitation was one of the reasons why most early applications of MPC involved systems with slow dynamics, such as chemical processes [54, 65]. Motivated by

This article is dedicated to the 70th birthday of Professor Hans Georg Bock, whose seminal contributions in the field of structure-exploiting real-time optimization algorithms inspired most of the reviewed developments.

✉ Dimitris Kouzoupis
dimitris.kouzoupis@imtek.uni-freiburg.de

1 Georges-Köhler-Allee 102, DE-79110, Freiburg im Breisgau, Germany

the improved control performance and the convenience in formulating the control objective and constraints, researchers in the field aimed at applying MPC on faster dynamical systems.

One popular approach to reduce the computational cost associated with an MPC controller is to use linear MPC, where typically the objective function is quadratic, while the system dynamics and the constraints on states and controls are linear [69]. Under these assumptions, the underlying optimization problems are quadratic programs (QP), parametric in the initial state. The control law is piecewise affine, defined on so-called *critical regions*, inside which the active set of the optimal solution is constant. The affine dependence of the control on the parameter allows one to precompute the feedback law for every point in the feasible parameter space, reducing the computational cost of the controller to a search in a lookup table for the right critical region [6]. However, this *explicit* approach becomes quickly intractable as the problem size, or more precisely the number of active set combinations, increases and iterative schemes are necessary to solve higher dimensional problems. A popular example is the online active set strategy `qpOASES`, which introduces a homotopy to traverse the critical regions, moving between QP solutions for different parameter values [20, 21]. First-order methods are also widely used schemes in linear MPC applications, due to the simplicity of their algorithmic schemes and the existence of practical complexity bounds when only input constraints are present [58, 71]. For input- and state-constrained problems, dual formulations have been proposed in the literature [61, 70], often exploiting the fact that the QP matrices remain constant, such that preconditioning can be applied to improve convergence [35]. The simplicity of first-order methods also motivated the development of several code generation tools [4, 79, 90].

In contrast to linear MPC, nonlinear MPC (NMPC) can directly handle any nonlinear (e.g., economic [13]) objective functions, nonlinear system dynamics, and general path constraints at the cost of more computationally expensive optimization problems. These attractive properties inspired a great progress in the field which led to a continuously shrinking gap in computational effort between linear and nonlinear MPC algorithms [16, 38, 47]. Since most dynamical systems are represented by ordinary differential equations or differential-algebraic equations, it is rather common in NMPC to have problem formulations in continuous time. A continuous time optimal control problem (OCP) lives in an infinite dimensional space and typically a parametrization is used to convert it into a nonlinear program (NLP). Methods that first discretize the OCP and then solve the finite dimensional NLP are called *direct methods* [8, 11]. Two popular families of algorithms to solve an NLP are interior point (IP) methods and sequential quadratic programming (SQP) [59, 64, 86]. In order to bring NMPC closer to applications involving fast dynamical processes, several real-time variants of SQP and IP methods have been proposed in the literature, aiming at finding only approximate solutions to the NLP [15, 55, 60]. For instance, the real-time iteration (RTI) scheme performs merely one SQP iteration per time step and divides the calculations into a preparation and a feedback phase, in order to apply the control input as quickly as possible after the state measurement is taken. If the sampling time is short enough, this approach is shown to closely approximate the optimal closed loop performance of a fully converged NMPC scheme [14, 15].

Considering that in most linear and many nonlinear MPC controllers, structured QP subproblems need to be solved at each time step, tailored algorithms and efficient software implementations become crucial for bridging the gap in computational overhead between optimization-based and conventional controllers. Over the past years, a great progress has been made in the field of embedded quadratic programming, revealing various ways to exploit the underlying problem structure in tailored interior point or active set-based methods. Some of the major contributions known to the authors are listed in chronological order in Table 1. The ones relevant to this paper are further discussed in Sections 2 and 3.

Summary and Contributions The aim of this paper is on the one hand to summarize many major theoretical and practical contributions in the field of quadratic programming for NMPC and on the other hand, to provide a consistent benchmark between some of the most popular embedded QP solvers. The code of this benchmark has been made publicly available and interested readers can either conduct further experiments using the existing, scalable example or compare the performance of the solvers on their own applications. Section 2 summarizes some of the most important algorithmic developments in the field of NMPC, focusing on the numerical schemes that are relevant to this paper. Section 3 discusses the different approaches that have been proposed for solving the quadratic subproblems. Section 4 presents the benchmark example that is used in Section 5 to compare the selected QP solvers. Section 6 concludes the paper and outlines future directions.

2 Algorithms for Nonlinear Model Predictive Control

In this section, we discuss some important developments in numerical methods for NMPC, focusing on direct approaches and especially on *direct multiple shooting*, which has drawn significant attention over the past decades [11, 76].

In direct multiple shooting, the control inputs at each discretization interval are parametrized by a finite set of parameters, usually piecewise constant values, and the state variables are represented by the solutions to initial value problems with timespan typically equal to one sampling time. Continuity constraints ensure that the states at the end of one interval are equal to the states at the beginning of the next, when the optimization algorithm has converged. The discrete time OCP arising from a direct multiple shooting parametrization has the form:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=0}^{N-1} L(x_k, u_k) + E(x_N) && \text{(1a)}\\
\text{subject to} \quad & x_0 = \hat{x}_0 && \text{(1b)}\\
& x_{k+1} = f(x_k, u_k), \qquad k = 0, \dots, N-1, && \text{(1c)}\\
& h(x_k, u_k) \leq 0, \qquad k = 0, \dots, N-1, && \text{(1d)}\\
& h_N(x_N) \leq 0, && \text{(1e)}
\end{aligned}$$

where $x \in \mathbb{R}^{(N+1)n_x}$ and $u \in \mathbb{R}^{Nn_u}$ denote the states and controls over the prediction horizon N. The nonlinear functions $L(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}$ and $E(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}$ represent stage and terminal cost respectively, while $h(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_c}$ and $h_N(\cdot) : \mathbb{R}^{n_x} \to \mathbb{R}^{n_{\tilde{c}}}$ are the nonlinear path constraints. Finally, the functions $f(\cdot) : \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \to \mathbb{R}^{n_x}$ represent the numerical simulation of the system dynamics, i.e., the solution to an initial value problem.

Two of the most common families of methods to solve (1) are IP and SQP algorithms. In an interior point framework, the non-smooth complementarity condition in the Karush–Kuhn–Tucker (KKT) system of (1) is replaced by a smooth nonlinear approximation. Following a different approach, SQP methods linearize the inequality constraints (1d)–(1e) such that finding a point that satisfies the KKT optimality conditions becomes equivalent to solving a QP. In both cases, the special optimal control structure of the problem ought

Table 1 Selected publications in the field of quadratic programming for optimal control

Year	Authors	Citations	Contribution
1984	Bock et al.	[11]	Multiple shooting algorithm with $\mathcal{O}(N^3)$ condensing, implemented in `MUSCOD`.
1984	Glad et al.	[36]	Active set method using a Riccati recursion for the factorization of the KKT matrix.
1991	Wright	[85]	Primal-dual interior point method exploiting the fixed bandwidth of the KKT matrix.
1991	Wright	[84]	Parallelizable factorization of banded matrices.
1994	Steinbach	[76]	Primal-dual interior point method with block sparse structure exploitation.
1994	Schmid et al.	[74]	`QPKWIK`: A dual active set method tailored to reduced Hessian SQP.
1998	Rao et al.	[68]	Interior point method using a Riccati recursion for the solution of the KKT system.
1999	Leineweber	[51–53]	`MUSCOD-II` with $\mathcal{O}(N^3)$ condensing.
2006	Bartlett et al.	[5]	`QPSchur`: A dual active set, Schur complement method for large scale quadratic programming.
2008	Ferreau et al.	[20, 21]	`qpOASES`: An online active set strategy.
2010	Wang et al.	[81]	Implementation of a primal interior point method tailored to MPC.
2011	Houska et al.	[40]	`ACADO Toolkit` with $\mathcal{O}(N^3)$ condensing and `qpOASES`.
2011	Ullmann	[79]	`FiOrdOs`: Code generation of primal and dual fast gradient methods.
2011	Patrinos et al.	[63]	A piecewise smooth Newton method for large-scale MPC.
2012	Kirches et al.	[45]	Complementary condensing.
2012	Mattingley et al.	[56]	`CVXGEN`: Interior point-based code generator for embedded convex optimization.
2012	Domahidi et al.	[19]	`FORCES`: Code generation of primal-dual interior point methods for optimal control.
2012	Frison	[25]	Condensing algorithm with $\mathcal{O}(N^2)$ complexity.
2012	Axehill et al.	[3]	$\mathcal{O}(N^2)$ condensing scheme based on Riccati recursion for direct calculation of factorized dense Hessian.
2012	Patrinos et al.	[61, 62]	Accelerated dual gradient-projection algorithm.
2013	Di Cairano et al.	[12]	Projection-free parallel quadratic programming for linear MPC.
2014	Frasch et al.	[23]	`qpDUNES`: A dual Newton strategy.
2014	Frison et al.	[30]	`HPMPC`: A high-performance interior point method.
2014	Giselsson	[35]	Preconditioned dual fast gradient methods.
2015	Axehill	[2]	Partial condensing.
2016	Frison et al.	[28]	$\mathcal{O}(N^2)$ condensing algorithm tailored to moving horizon estimation and partial condensing.
2017	Stellato et al.	[4, 78]	`OSQP`: Operator splitting solver with code generation functionality.
2017	Frison et al.	[29]	`BLASFEO`: High-performance linear algebra library.

to be exploited in an efficient implementation. For an SQP method, this implies that tailored algorithms to solve QP subproblems with block-banded structure are essential. An overview of such algorithms is given in Section 3. One advantage of SQP over IP methods is *warm-starting*, i.e., their ability to use a good initial guess of the optimal solution to accelerate convergence. Such an initial guess is readily available in an MPC framework from the solution of the previous problem. IP methods on the other hand typically need to follow the so-called central path, which makes the impact of warm-starting less effective. Some remedies to this problem can be found in [37, 75, 88].

Bock and Plitt in [11] propose to solve (1) with an SQP method, using line search to ensure global convergence and partitioned, high rank updates of the Hessian approximation to preserve the problem structure. The sparse QPs are first transformed into dense ones using *condensing*, an algorithm that eliminates the state variables from the optimization problem, and then solved with a dense QP solver. The first implementation of this scheme was part of the proprietary package `MUSCOD`, which was later succeeded by `MUSCOD-II` [50, 52, 53]. Among the used QP solvers were `QPSOL` and its successor `QPOPT` [34]. In order to reduce the feedback delay in NMPC, Diehl et al. in [15] propose to solve the NLP (1) only approximately. Based on the observation that the state of the system is continuously changing while the controller is calculating the next optimal input, the real-time iteration (RTI) scheme minimizes the time between state measurement and control feedback by performing only one SQP iteration per time step and splitting the calculations into a preparation and a feedback phase. All operations that are independent of the state measurement belong to the preparation step. The feedback step comprises the *initial value embedding* and the solution of the QP, which are typically performed faster. One way to interpret this numerical scheme is *iterating while the problem changes* and if implemented with care, it does not compromise closed loop performance [14]. No globalization strategy is used in this context, under the assumption that full steps are sufficient to guarantee local convergence if the sampling time is short enough and the problems are initialized properly. Convergence and stability guarantees for the simplified setup with only equality constraints are provided in [17]. To reduce the computational load even further, the multi-level RTI scheme divides the operations into levels that are executed with different frequencies [10, 24, 82]. A first implementation of the RTI scheme was part of `MUSCOD-II`. Later on, it was reimplemented in the open-source software package `ACADO Toolkit` [40]. Upon its release, `ACADO` provided both direct single and multiple shooting as parametrization schemes, several SQP-based algorithms and condensing in combination with `qpOASES` for the solution of the dense QPs. Targeting embedded applications with sampling times in the microsecond range, a code generation framework became part of `ACADO` soon after its release [41, 66]. It consisted of an RTI scheme with the absolute minimum necessary components to solve a nonlinear OCP with a least squares objective using the Gauss–Newton Hessian approximation [9]. Code generation was used, among other places, in differentiation routines, in numerical simulation, and in the condensing algorithm, yielding a significant performance improvement over the original C++ implementation.

Other real-time algorithms for NMPC are—among many—the Newton-type controller by Li and Biegler [55], also performing one SQP iteration with a Gauss–Newton Hessian per time step but using line search and a single shooting formulation [73]; the advanced step controller by Zavala and Biegler [89], which performs IP iterations until convergence but compensates for the delay by using a predicted initial condition and applying a tangential predictor to correct for the difference from the actual state measurement; and the continuation/GMRES method by Ohtsuka [60], performing one Newton-type iteration with exact Hessian on the single shooting formulation and treating inequalities in an IP-like fashion with a fixed barrier parameter. Several approaches have also appeared in the literature that solve the arising NLPs in a distributed manner, such as the augmented Lagrangian-based method of [42] and the decomposition algorithm in [39]. In NMPC with only input constraints, simpler numerical schemes such as the the projected gradient method in [44], the proximal method of [77] and the projected Newton methods in [7, 32] are also possible alternatives.

3 Solving the Quadratic Subproblems

The linear-quadratic subproblems arising in an NMPC scheme based on sequential quadratic programming can be written in the form:

$$\begin{aligned}
\underset{x,u}{\text{minimize}} \quad & \sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} x_k \\ u_k \\ 1 \end{bmatrix}^\top \begin{bmatrix} Q_k & S_k & q_k \\ S_k^\top & R_k & r_k \\ q_k^\top & r_k^\top & 0 \end{bmatrix} \begin{bmatrix} x_k \\ u_k \\ 1 \end{bmatrix} + \begin{bmatrix} x_N \\ 1 \end{bmatrix}^\top \begin{bmatrix} Q_N & q_N \\ q_N^\top & 0 \end{bmatrix} \begin{bmatrix} x_N \\ 1 \end{bmatrix} && \text{(2a)} \\
\text{subject to} \quad & x_0 = \hat{x}_0, && \text{(2b)} \\
& x_{k+1} = A_k x_k + B_k u_k + b_k, \qquad k = 0, \ldots, N-1, && \text{(2c)} \\
& \underline{d}_k \leq C_k x_k + D_k u_k \leq \bar{d}_k, \qquad k = 0, \ldots, N-1, && \text{(2d)} \\
& \underline{d}_N \leq D_N x_N \leq \bar{d}_N, && \text{(2e)}
\end{aligned}$$

which is a sparse, structured QP with the dynamic constraints in (2c) imposing the only coupling between variables of consecutive stages. General purpose sparse QP solvers such as `OOQP` [33] can be used to solve (2) and they typically perform significantly better than dense solvers that assume that all entries in the problem are non-zero. However, the specific sparsity pattern of optimal control problems (2a)–(2e) should be exploited in order to maximize performance. Three schemes have been proposed in the literature to solve block-banded optimization problems in the form of (2), combining structure exploitation with operations on small-scale, dense matrices. This is the topic of the following sections.

Remark 1 (Banded structure exploitation in single shooting) The band structure-exploiting methods described in this section can also be used within a single shooting framework by simply overwriting the states of the QP solution with a nonlinear forward simulation [69].

3.1 The Condensing Approach

A closer look at the equality constraints (2b)–(2c) reveals that the state variables x are uniquely determined by the input variables u and the initial state $\hat{x}_0$. Therefore, one solution approach is to eliminate the states x from the optimization problem and use a dense solver, such as `qpOASES`, to solve the smaller scale QP that has the form:

$$\begin{aligned}
\underset{y}{\text{minimize}} \quad & \frac{1}{2} y^\top H_\text{d}\, y + y^\top g_\text{d}, && \text{(3a)} \\
\text{subject to} \quad & \underline{d}_\text{d} \leq C_\text{d}\, y \leq \bar{d}_\text{d}. && \text{(3b)}
\end{aligned}$$

The elimination scheme that transforms (2) into (3) is often referred to as *condensing algorithm* [11].

Within an NMPC framework, the condensing algorithm needs to be executed at each time step. This implies that, in particular for large N, the operation consumes a significant part of the overall execution time. For a long period of time, the complexity of the condensing algorithm was thought to be cubic in the prediction horizon N [41, 80]. However, carefully exploiting the problem structure during the elimination of the state variables can lead to algorithms that scale quadratically in N, as independently discovered by the authors in [1, 25]. This improvement can significantly reduce the computation time of the NMPC scheme, as shown in the results of Section 5. However, the asymptotic complexity of the complete solution approach (i.e., condensing and a prescribed number of iterations for the dense QP) remains cubic in N if an active set or interior point method is used, due to the factorization of the dense Hessian matrix. This fact can be overcome with the condensing algorithms proposed in [3, 27] that directly construct the Cholesky factor of H_{d} without building the matrix itself. If these condensing algorithms are combined with an active set method that performs updates of the KKT matrix factor at each iteration (with complexity that is quadratic in the number of variables), the resulting scheme to solve (2) will have an asymptotic complexity that scales only quadratically in N. Yet another condensing algorithm that outperforms its competitors when the initial state is free, i.e., when constraint (2b) is not present, has been recently proposed in [28]. For a detailed comparison of condensing algorithms, the reader is referred to [31].

Remark 2 (Complementary condensing) In optimization problems where the number of controls is much larger than the number of states, such as in convexification approaches for mixed-integer OCPs [46, 72], the reduction in problem size that the conventional condensing algorithm achieves is marginal. To alleviate this problem, a complementary condensing approach that is essentially a structure-exploiting factorization of the block sparse KKT system of (2) has been proposed in [45].

3.2 Exploiting the Block-Banded Structure

Although condensing is a competitive approach for relatively small N, it is less suitable for problems with long horizons. This drawback motivated the development of solution algorithms that exploit the optimal control structure of the QP in a different way.

When no inequalities are present in (2), the optimal solution can be calculated using a Riccati recursion, which is related to dynamic programming and has a computational cost that scales only linearly in N. This approach had undoubtedly a major influence also in constrained optimization. In the early work by Glad and Johnson [36], an active set method is proposed that uses a Riccati recursion to efficiently solve the KKT system associated with (2) for different guesses of the optimal active set. Similar results followed in the field of interior point methods. Wright in [85] carefully restructures the KKT matrix into a block-banded form and factorizes it efficiently using a banded factorization with fixed bandwidth. The parallelization of this procedure can be achieved by means of a cyclic reduction technique proposed by the same author in [84]. It requires more floating point operations in total, but spread over different processors. Later work that fully exploits the block-banded structure of the problem by means of Schur complements or Riccati recursions followed by Steinbach and Rao et al. in [68, 76].

The performance and robustness of interior point methods motivated also the development of several software tools, with some representative examples discussed in Section 5. An efficient implementation of an interior point algorithm that follows the Riccati recursion approach of Rao et al. [68] was proposed by Frison et al. in [30]. Wang et al. in [81] and Domahidi et al. in [19] use instead a Schur complement method within an interior point algorithm to form and factorize the KKT matrix in the so-called *normal equations form*. The factorization step is based on a banded Cholesky factorization procedure. The same block tridiagonal matrix appears in the active set-based method by Frasch et al. in [22], as the Hessian of a dual problem formulation. The authors therein propose a reverse Cholesky factorization with the same complexity in order to maximize reuse of results between iterations, as discussed in Section 5.

If the matrices A_k and B_k in (2c) are assumed to be dense, using direct sparse solvers, treating the block sparse matrices as banded matrices or fully exploiting the special block structure yields algorithms with the same asymptotic complexity, namely linear in N and cubic in the number of stage variables. However, in practice, the performance gap between such methods can be more than one order of magnitude, as for instance reported in [26].

3.3 Partial Condensing

A natural extension of the previous two approaches is to combine them in one by applying the condensing algorithm on consecutive blocks of size $M \leq N$. This method, called partial (or block) condensing, allows one to form optimization problems with different levels of sparsity [2]. The resulting QP, with a fictitious horizon $\tilde{N} < N$ and a larger number of control variables per stage, can be solved using the algorithms of the previous section.

As an example, consider a QP (2) with horizon length $N = 6$ and let us divide the state and input vectors such that the variables of $M = 3$ consecutive stages are grouped together. Eliminating all but the first state on each block yields:

$$\tilde{u} = \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ \hline u_3 \\ u_4 \\ u_5 \end{bmatrix} \doteq \begin{bmatrix} \tilde{u}_0 \\ \tilde{u}_1 \end{bmatrix}, \quad \tilde{x} = \begin{bmatrix} x_0 \\ x_3 \\ x_6 \end{bmatrix} \doteq \begin{bmatrix} \tilde{x}_0 \\ \tilde{x}_1 \\ \tilde{x}_2 \end{bmatrix},$$

with $\tilde{x}_k \in \mathbb{R}^{n_x}$ and $\tilde{u}_k \in \mathbb{R}^{Mn_u}$. These vectors can be seen as the state and controls of a new block-banded QP (2) with horizon length $\tilde{N} = N/M = 2$. The matrices and vectors of the QP are replaced by their bar quantities, calculated by a partial condensing algorithm. Note that the condensing of each block can be performed in parallel. Moreover, if the QP solver supports varying dimensions of states and controls per time stage, the block size M does not have to be a proper divisor of N, allowing more freedom in the tuning process.

Initially, the main motivation behind partial condensing was the reduction of floating point operations inside the QP solver by changing the problem dimensions [2]. However, additional advantages include the improved efficiency of linear algebra routines (the partial condensing reformulation replaces many linear algebra operations on small matrices with fewer linear algebra operations on larger matrices [28]) and the acceleration of convergence for some algorithmic schemes [49].

As in standard condensing, partial condensing needs to be performed at each sampling time for NMPC. However, in partial condensing, the initial state of each but the first block is a free optimization variable. This is a key difference with respect to the full condensing approach of Section 3.1, where the initial state constraint (2b) can be used to eliminate x_0 from the optimization variables, in order to reduce the computational complexity of the algorithm. In this context, the best-performing condensing algorithm is the one proposed in [28]. An efficient implementation of this scheme can be found in the open-source project HPIPM [94].

4 The Nonlinear Hanging Chain Problem

In this section, we present the benchmark problem that will be used to compare the computational performance of recently developed software in the field of embedded quadratic programming. The chosen problem is scalable in the number of states, which allows one to conveniently investigate the behavior of the solvers in problems with different dimensions.

The controlled system is a chain of n_m masses connected by springs, as already used in [20, 80, 83]. The mass at the one end of the chain is fixed, while the mass on the other end is actuated. The model has $n_x = 6(n_\mathrm{m} - 2) + 3$ states, corresponding to the three-dimensional positions and velocities of the intermediate masses and the position of the free mass at the one end, and $n_u = 3$ controls for the velocities of the actuated mass at that end.

Initially, the chain rests in an equilibrium position with the y and z coordinates of the two ends coinciding, as shown in Fig. 1. A wall near this equilibrium position, introduced in [20], imposes state bounds on the y position of each free mass. The control scenario starts with a strong perturbation to the chain by moving its actuated end with a fixed velocity of $[-1, 1, 1]$ for five sampling periods. After this time, the NMPC controller is taking over with the task to return the chain to its initial position. Control values are bounded between -1 and 1.

The continuous time OCP is parametrized using direct multiple shooting with piecewise constant controls on intervals of $T_s = 0.2$ s. For the integration and linearization of the system dynamics, two steps of the 2nd-order Gauss collocation-based integrator with tailored sensitivity propagation are used [67]. The same integration scheme is used for the simulation model in the closed loop experiments. The objective function is quadratic, with diagonal weights, penalizing the distance of states and controls from their initial values. This setup allows the use of a Gauss–Newton Hessian approximation, which is applicable to objective functions in nonlinear least squares form [9]. The NMPC controller is based on the RTI scheme, as implemented in the `ACADO Code Generation` tool [41].

A series of closed loop simulations for different values of the prediction horizon N is executed for the considered QP solvers and the worst case computation time of each simulation appears in the plots. The timings refer to condensing or partial condensing, if applicable, and QP solution. The returned primal optimal solution of every subproblem is validated against `qpOASES` with condensing, ensuring that the infinity norm of the absolute difference is smaller than 10^{-5}. All simulations run on a Dell XPS 13 9360 equipped with an Intel i7-7560U CPU running at 2.30 GHz and with the Turbo Boost feature disabled to avoid clock frequency fluctuations. The code used to produce the results of the following section is publicly available on Github [93] (version 0.1).

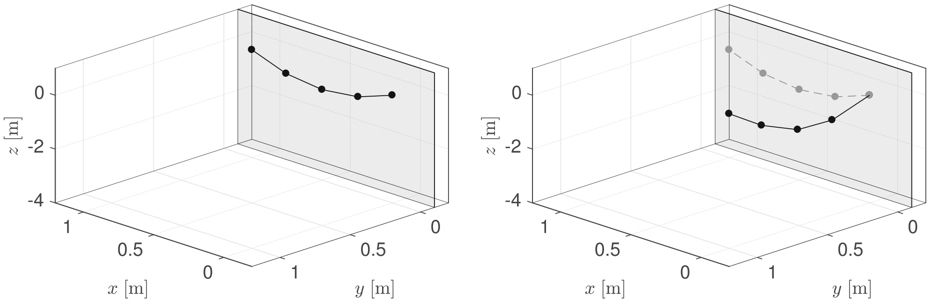

Fig. 1 The hanging chain in equilibrium position (left) and after the perturbation (right)

5 Benchmark of Recent Software Developments

The aim of this section is to summarize and benchmark several software developments of the past decade in the field of embedded quadratic programming for NMPC. Note that this overview is far from complete, as it is based on QP solvers with an interface to the `ACADO Code Generation` tool.

5.1 The Online Active Set Strategy qpOASES with Condensing

The open-source software `qpOASES` implements an active set method to solve the dense QP (3), with attractive properties for MPC applications [20, 21]. It is written in C++ and was one of the first QP solvers coupled to the `ACADO Code Generation` tool.

The online active set strategy exploits the fact that a parametric QP can be solved for any feasible value of its parameter starting from the solution with respect to another value and following a homotopy path on the parameter space. In the context of linear MPC, the matrices of the QPs remain constant and the vectors depend only on the initial state measurement. Therefore, the parameter space is defined by all values of $\hat{x}_0$ for which QP (3) has a non-empty feasible set. If the sampling time is short enough, few active set iterations are typically required to recover the solution to the new problem. Moreover, matrix factorizations can be reused not only within the active set iterations, but also across successive QP instances, reducing the computational cost significantly. Another advantage of the online active set strategy over standard primal or dual active set methods is the clear interpretation of its solution, if the algorithm is stopped prematurely. More precisely, all intermediate iterates retain primal and dual feasibility with respect to their corresponding parameter value [20]. In an NMPC framework, where the matrices in (3) are no longer constant, the algorithm requires a factorization step for every new problem followed by the construction of an auxiliary QP with modified vectors in order to follow the same homotopy strategy. Both the factorization of the condensed Hessian in `qpOASES` and the first code-generated condensing algorithm in `ACADO` had a cubic complexity in the horizon length. The development of more efficient condensing algorithms with quadratic complexity in N improved the performance of this solution scheme significantly, despite of the fact that the asymptotic complexity remained cubic in N. This is demonstrated in Fig. 2, where a comparison of the computation time for the two schemes as a function of N is shown both in linear and logarithmic scale, for a chain of 3, 4, and 5 masses respectively. The reported timings refer to worst case computation time in closed loop, as discussed in Section 4.

Remark 3 (A sparse variant of `qpOASES`) For the simulations of this paper, the dense variant of `qpOASES` is used from `ACADO` in combination with code-generated condensing algorithms. For a sparse variant of the online active set strategy, whose complexity per iteration is between linear and quadratic in N, the reader is referred to [43].

5.2 The Code-Generated Interior Point Solvers of FORCES

One year after the RTI scheme based on condensing and `qpOASES` was published, the code generation framework `FORCES` appeared [19], implementing primal-dual interior point methods tailored to the block-banded QP (2). Upon its release, the code generator could be used free of charge. Nowadays, `FORCES Pro` is a commercial software developed and supported by the company embotech AG [18]. The convex QP solver is extended to support more algorithms and an NLP solver for optimal control is also available [87].

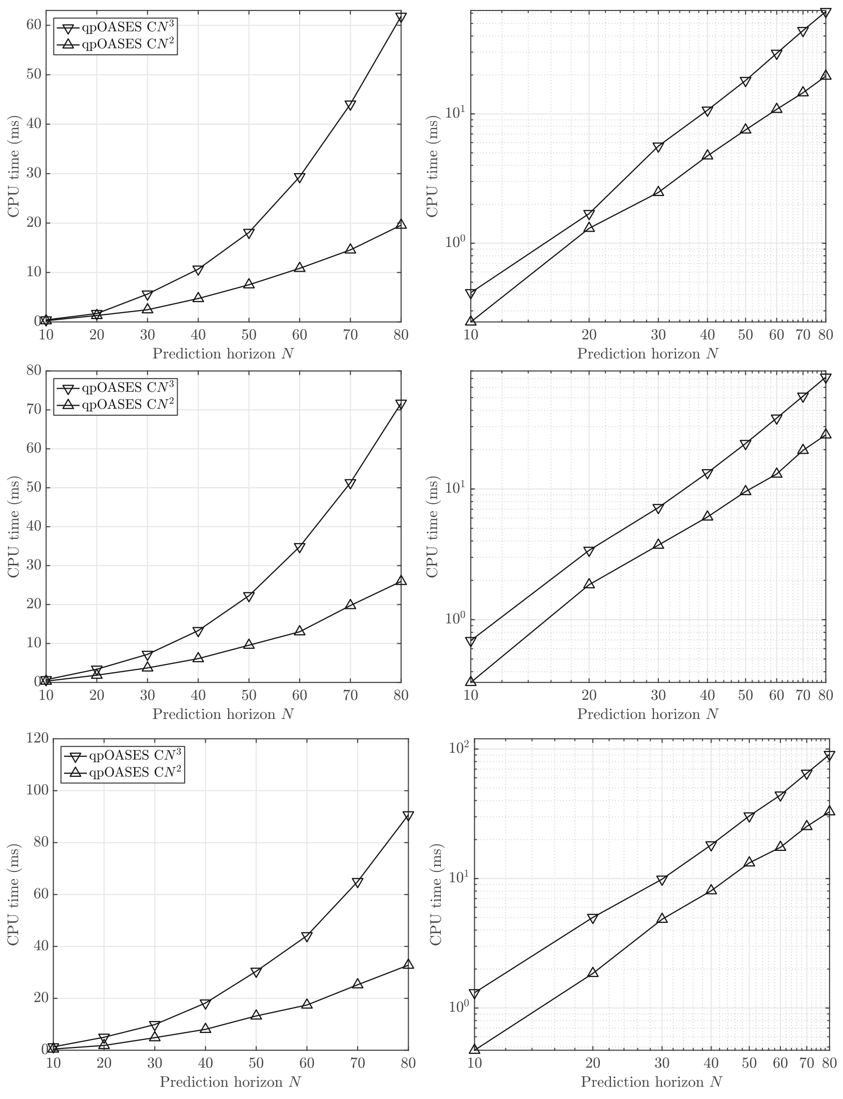

Fig. 2 Performance of `qpOASES` with the two different condensing algorithms for a hanging chain of 3 ($n_x = 9$), 4 ($n_x = 15$), and 5 ($n_x = 21$) masses respectively in linear (left) and logarithmic (right) scale

An advantage of interior point over active set methods is that they typically exhibit lower fluctuation in the number of iterations required to solve different QP instances. The most computationally intensive operation in an interior point algorithm is the computation of the search direction, which can be seen as a Newton step on the nonlinear root finding problem defined by the relaxed KKT conditions [59]. Slack variables and Lagrange multipliers of inequality constraints are usually eliminated by properly updating Hessian and gradient. The resulting symmetric, indefinite system of equations is equivalent to the KKT conditions of an equality constrained QP, often referred to as the *augmented system*. `FORCES` implements a Schur complement method to solve the augmented system. As a first algorithmic step, the Schur complement is applied independently at each stage of the QP and a system of linear equations with a symmetric, positive definite block tridiagonal matrix is formed. This system of linear equations is then solved using a block-wise Cholesky factorization, with a computational complexity that is linear in N and cubic in the number of stage variables.

In order to use `FORCES`, the optimization problem is formed in MATLAB or Python and a request is sent to a server to generate ANSI C code and download the compiled binaries of the QP solver. Problem data can be fixed at code generation time or left as a parameter to be provided at run time. Level 3 `BLAS`- and `LAPACK`-like routines are coded using triple loops, where the size of loops is fixed in the code, giving the compiler the chance to perform optimizations like loop unrolling. Due to the fact that disclosing the computational performance of `FORCES` in this example would violate the terms of the academic license agreement, the interested readers can either reproduce the results using the publicly available code in [93] and their own `FORCES Pro` license or look into [80] for an almost identical comparison on a different computer architecture.

5.3 The Dual Newton Strategy qpDUNES

Another active set method tailored to block-banded QPs in the form of (2) is the so-called dual Newton strategy, as implemented in the open-source software `qpDUNES` [22].

The main idea of the algorithm is to introduce Lagrange multipliers for the coupling constraints in (2c) and solve the resulting dual problem with Newton's method. The dual function is concave, piecewise quadratic and once continuously differentiable if the Hessian blocks are positive definite. Line search is used to ensure global convergence. The construction of the Newton system at each iteration requires the solution of $N + 1$ small-scale QPs (since the variables of each stage are now decoupled) and the calculation of the dual gradient and Hessian. These operations can be performed in parallel to a large extent and typically reuse computations from the previous iteration. More precisely, parts of the dual system that correspond to a stage QP without an active set change remain constant. The Hessian of the dual problem has a block tridiagonal structure, which is exploited by means of a banded Cholesky factorization, similarly to `FORCES`. Since the factorization only needs to start from the Hessian block where the first active set change occurs, a reverse Cholesky factorization is used, based on the assumption that most active set changes appear at the beginning of the prediction horizon.

If the inequality constraints in (2d)–(2e) contain only bounds on the stage variables and the Hessian of the QP is diagonal, the variables of the stage QPs are completely decoupled and their solution can be calculated analytically, with a simple clipping operation. In any other case, the QPs are solved numerically using `qpOASES`, which has several benefits over other QP solvers in this context. For instance, the Hessian matrix of the stage QPs needs to be factorized only once and the null space of the active constraints which is calculated inside `qpOASES` can be reused when building the dual Hessian. `qpDUNES` is written in C, including a C++ layer to communicate with `qpOASES`. Similarly to `qpOASES`, the performance of `qpDUNES` typically improves with warm-starting. A main advantage of `qpDUNES` is its ability to perform multiple active set changes simultaneously. On the other hand, if the algorithm is stopped prematurely, its solution is not feasible nor optimal for a neighboring MPC problem.

The computation time of `qpDUNES` with clipping is shown in Fig. 3. As expected, there is a crossover point after which the sparse solution approach becomes more efficient than the combination of condensing with a dense QP solver. This crossover point also depends on the number of states. The more states are eliminated, the more prominent the benefits of condensing are.

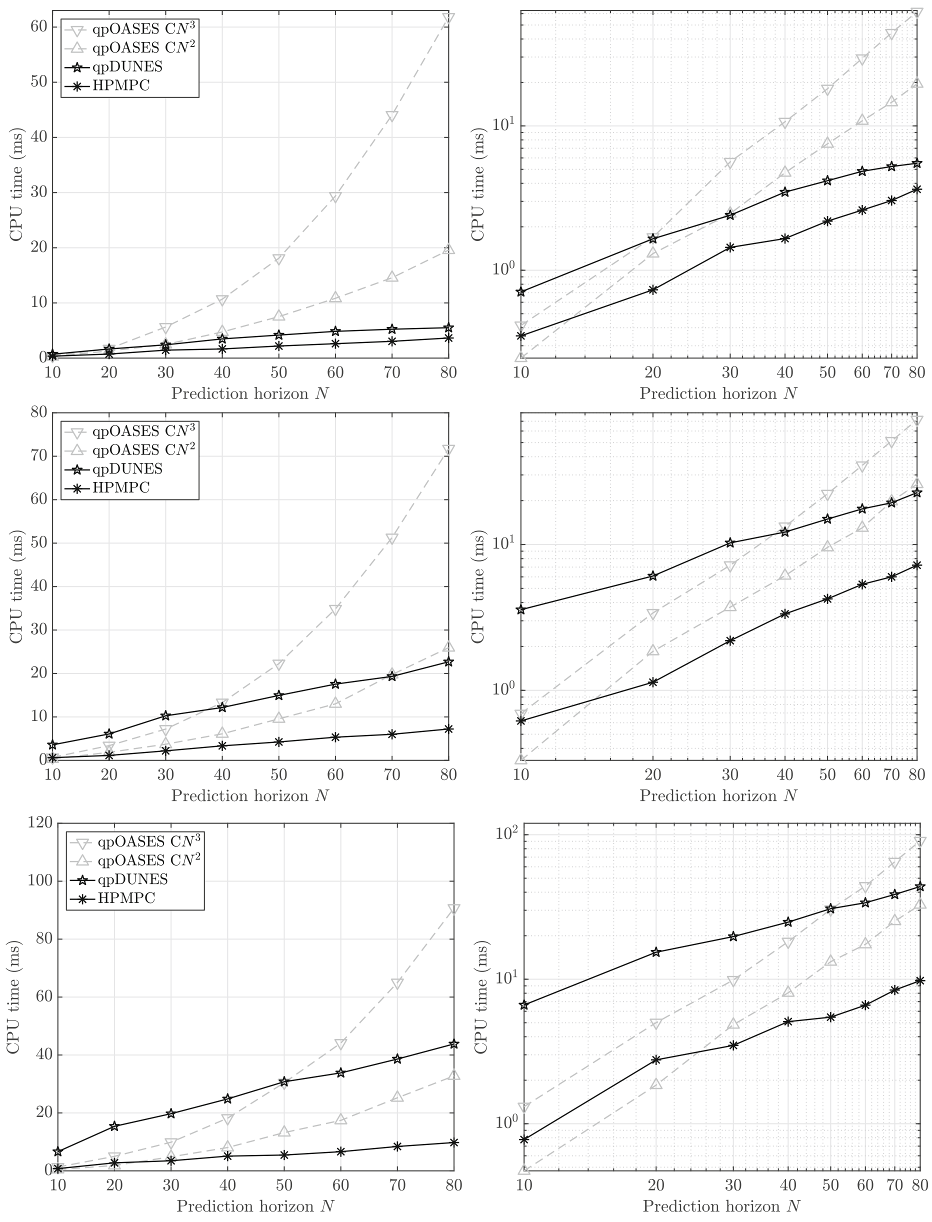

Fig. 3 Performance of the banded structure-exploiting solvers `qpDUNES` and `HPMPC` for a hanging chain of 3 ($n_x = 9$), 4 ($n_x = 15$), and 5 ($n_x = 21$) masses respectively in linear (left) and logarithmic (right) scale

5.4 The High-Performance Interior Point Method HPMPC

The open-source software `HPMPC` [30, 95] implements a primal-dual interior point method also tailored to optimal control structured QPs. The main feature that distinguishes it from other solvers is the use of efficient basic linear algebra routines, which are optimized for small- to medium-scale matrices.

The search direction in HPMPC is computed using a backward Riccati recursion. This recursion has the same asymptotic complexity as the Schur complement method implemented in FORCES. However, the Riccati recursion has a lower flop count in practice, especially if the Hessian is not diagonal or there exist general affine constraints in the QP. HPMPC does not employ code generation and it comes in the form of a C library. It comprises a set of linear algebra kernels implemented using intrinsics in order to leverage the vectorization capabilities of modern CPUs. Their aim is to provide a set of basic linear algebra routines optimized for small matrix sizes that are common in embedded optimization, similarly to what BLAS and LAPACK do for large matrix sizes that are common in high performance computing.

Figure 3 demonstrates that the high-performance linear algebra routines implemented in HPMPC yield a significant improvement over previously presented software. This enhanced performance inspired the development of the open-source library BLASFEO [29, 92], which makes the linear algebra routines of HPMPC accessible to other software developments in the field (see, e.g., treeQP [48, 96]).

5.5 Partial Condensing with qpDUNES and HPMPC

The performance of any banded structure-exploiting QP solver can be potentially improved when combined with the partial condensing approach of Section 3.3.

In the work of [49], qpDUNES was combined with a code-generated partial condensing algorithm provided by ACADO. A significant speedup in performance was achieved, mainly due to the fewer Newton iterations required for the algorithm to converge. The effect of partial condensing for qpDUNES in the current benchmark example is shown in Fig. 4 for $N = 60$. The reported speedup with respect to $M = 1$ refers to worst case computation time in closed loop, similarly to the previous section. When no partial condensing is applied, qpDUNES is used in combination with clipping. For $M > 1$, the stage QPs are solved with qpOASES, since the Hessian blocks are no longer diagonal and the state bounds are transformed into general constraints. This explains why the performance deteriorates in the transition from $M = 1$ to $M = 2$. Moreover, since qpDUNES only supports fixed state and control dimensions per stage, the block size can only be a proper divisor of N.

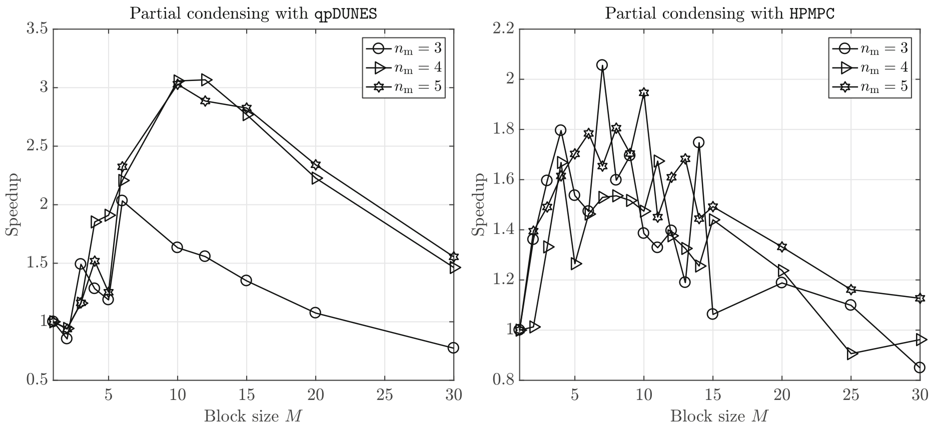

Fig. 4 Speedup of qpDUNES and HPMPC using partial condensing

The effect of partial condensing on HPMPC is mainly related to the flop reduction inside the interior point algorithm and the increase in dimensions of the dense sub-matrices that are passed to BLASFEO routines [28, 29]. Partial condensing achieves for this example a maximum speedup of about 2. Note that HPMPC uses its own partial condensing algorithm, based

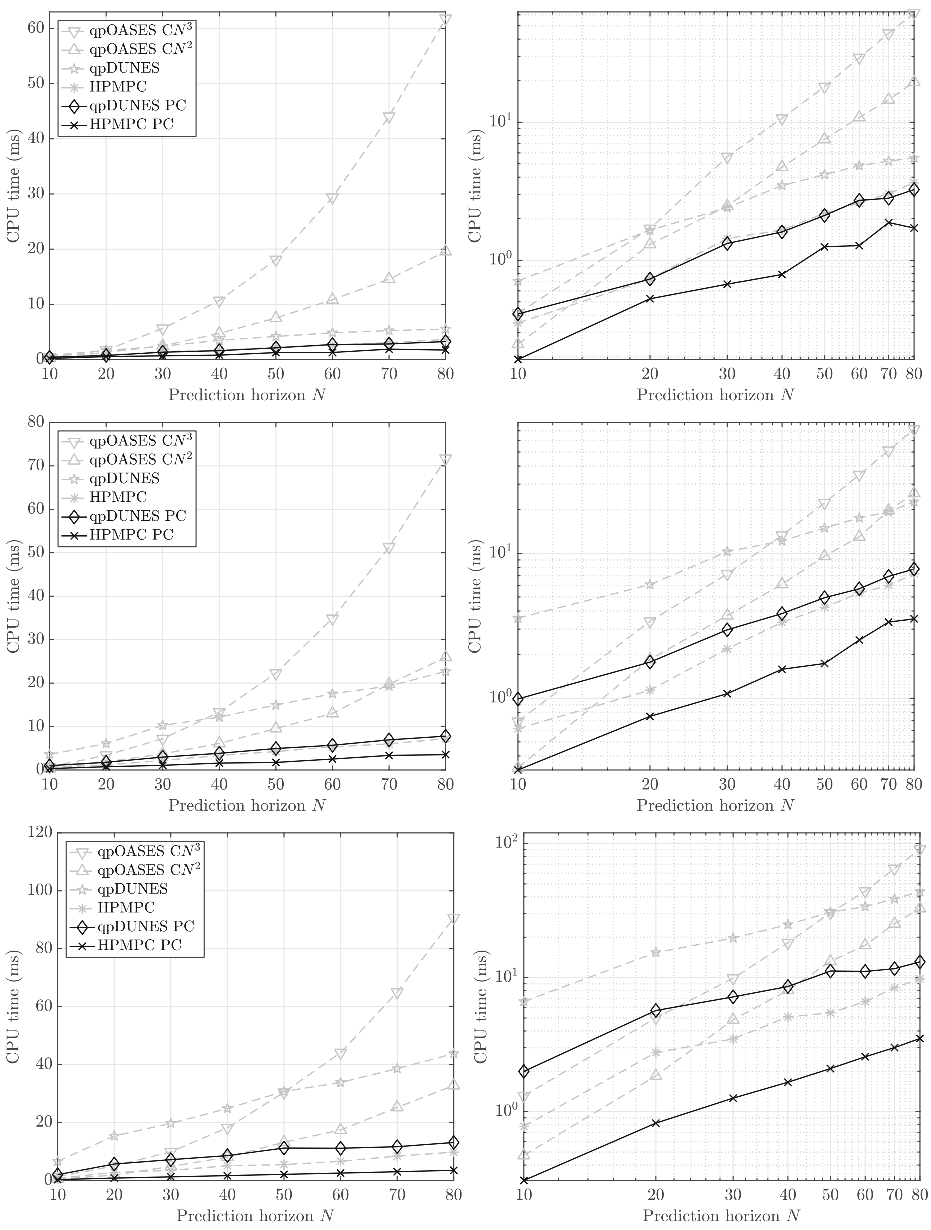

Fig. 5 Performance of banded structure-exploiting solvers qpDUNES and HPMPC with partial condensing ($M = 10$) for a hanging chain of 3 ($n_x = 9$), 4 ($n_x = 15$), and 5 ($n_x = 21$) masses respectively in linear (left) and logarithmic (right) scale

on BLASFEO. Moreover, since HPMPC supports varying number of variables per stage, the search for the optimal block size is made on a finer grid. A comparison of qpDUNES and HPMPC with partial condensing of block size 10^1 against the solvers of the previous sections is finally provided in Fig. 5.

6 Conclusion and Outlook

In this paper, we provided an overview on the latest algorithmic and software developments in the field of embedded QP solvers for NMPC, together with a consistent benchmark of selected software in the field. The numerical results indicate that the progress over the past few years brought improvements in computational performance of up to an order of magnitude.

Efficient dense linear algebra routines seem to be a powerful alternative to code generation techniques while at the same time they lead to cleaner and more maintainable software. With high performance, code maintainability, and modularity in mind, a successor of the ACADO Code Generation tool is currently being developed, under the name acados [91]. It replaces code-generated linear algebra routines with calls to BLASFEO functions and uses HPIPM [94] (a modular reimplementation and extension of HPMPC) to solve QPs or transform them, using condensing and partial condensing. Interfaces to third party software, such as the open-source QP solvers of Section 5, are also provided. The default algorithmic differentiation software is CasADi [1], but it can be replaced with any other tool that has the same functionality. In fact, the user can also directly provide self-written C code that evaluates the nonlinear functions and the necessary derivatives. acados is written in C and comes with interfaces to C++, Python, and MATLAB. A detailed comparison to its predecessor is subject of ongoing research.

Acknowledgements The authors are also grateful for the feedback of Per Rutquist and Yuning Jiang.

Funding Information This research was financially supported by the EU via FP7-ITN-TEMPO under grant 607 957 and H2020-ITN-AWESCO under grant 642 682, by the Federal Ministry for Economic Affairs and Energy (BMWi) via eco4wind under grant 0324125B, by DFG via Research Unit FOR under grant 2401, and by the State Ministry of Baden-Wuerttemberg for Sciences, Research and Arts under grant Az: 22-7533.-30-20/9/3.

References

1. Andersson, J.: A General-Purpose Software Framework for Dynamic Optimization. PhD thesis, KU Leuven, Leuven (2013)
2. Axehill, D.: Controlling the level of sparsity in MPC. Syst. Control Lett. **76**, 1–7 (2015)
3. Axehill, D., Morari, M.: An alternative use of the Riccati recursion for efficient optimization. Syst. Control Lett. **61**, 37–40 (2012)
4. Banjac, G., Stellato, B., Moehle, N., Goulart, P., Bemporad, A., Boyd, S.: Embedded code generation using the OSQP solver. In: Proceedings of the IEEE 56th Annual Conference on Decision and Control (CDC). Melboume, Australia (2017)
5. Bartlettand, R.A., Biegler, L.T.: QPSChur: A dual, active-set, Schur-complement method for large-scale and structured convex quadratic programming algorithm. Optim. Eng. **7**, 5–32 (2006)

[1]The value $M = 10$ might not achieve the best performance in Fig. 4 but it ensures that the block size is a proper divisor of N for all its values in the plot (required for qpDUNES).

6. Bemporad, A., Borrelli, F., Morari, M.: The explicit solution of constrained LP-based receding horizon control. In: Proceedings of the 39th IEEE Conference on Decision and Control (CDC). Sydney, Australia (1999)
7. Bertsekas, D.P.: Projected Newton methods for optimization problems with simple constraints. SIAM. J. Control Optim. **20**, 221–246 (1982)
8. Biegler, L.T.: Solution of dynamic optimization problems by successive quadratic programming and orthogonal collocation. Comput. Chem. Eng. **8**, 243–248 (1984)
9. Bock, H.G.: Recent advances in parameteridentification techniques for O.D.E. In: Deuflhard, P., Hairer, E. (eds.) Numerical Treatment of Inverse Problems in Differential and Integral Equations, pp. 95–121. Birkhäuser, Boston (1983)
10. Bock, H.G., Diehl, M., Kostina, E.A., Schlöder, J.P.: Constrained optimal feedback control of systems governed by large differential algebraic equations. In: Biegler, L.T. et al. (eds.) Real-Time and PDE-Constrained Optimization, pp. 3–22. SIAM, Philadelphia (2007)
11. Bock, H.G., Plitt, K.J.: A multiple shooting algorithm for direct solution of optimal control problems. In: Proceedings of the IFAC World Congress, pp. 242–247. Pergamon Press, Oxford (1984)
12. Di Cairano, S., Brand, M., Bortoff, S.A.: Projection-free parallel quadratic programming for linear model predictive control. Int. J. Control **86**, 1367–1385 (2013)
13. Diehl, M., Amrit, R., Rawlings, J.B.: A Lyapunov function for economic optimizing model predictive control. IEEE Trans. Autom. Control **56**, 703–707 (2011)
14. Diehl, M., Bock, H.G., Schlöder, J.P.: A real-time iteration scheme for nonlinear optimization in optimal feedback control. SIAM J. Control Optim. **43**, 1714–1736 (2005)
15. Diehl, M., Bock, H.G., Schlöder, J.P., Findeisen, R., Nagy, Z., Allgöwer, F.: Real-time optimization and nonlinear model predictive control of processes governed by differential-algebraic equations. J. Process Control **12**, 577–585 (2002)
16. Diehl, M., Ferreau, H.J., Haverbeke, N.: Efficient numerical methods for nonlinear MPC and moving horizon estimation. In: Magni, L., Raimondo, M.D., Allgöwer, F. (eds.) Nonlinear Model Predictive Control. Lecture Notes in Control and Information Sciences, vol. 384, pp. 391–417. Springer, Berlin (2009)
17. Diehl, M., Findeisen, R., Allgöwer, F.: A stabilizing real-time implementation of nonlinear model predictive control. In: Biegler, L.T., et al. (eds.) Real-Time and Online PDE-Constrained Optimization, pp. 23–52. SIAM (2007)
18. Domahidi, A., Perez, J.: FORCES professional. http://embotech.com/FORCES-Pro (2013)
19. Domahidi, A., Zgraggen, A., Zeilinger, M.N., Morari, M., Jones, C.N.: Efficient interior point methods for multistage problems arising in receding horizon control. In: Proceedings of the IEEE Conference on Decision and Control (CDC), pp. 668–674. Maui, HI (2012)
20. Ferreau, H.J., Bock, H.G., Diehl, M.: An online active set strategy to overcome the limitations of explicit MPC. Int. J. Robust Nonlinear Control **18**, 816–830 (2008)
21. Ferreau, H.J., Kirches, C., Potschka, A., Bock, H.G., Diehl, M.: qpOASES: A parametric active-set algorithm for quadratic programming. Math. Program. Comput. **6**, 327–363 (2014)
22. Frasch, J.V., Sager, S., Diehl, M.: A parallel quadratic programming method for dynamic optimization problems. Math. Program. Comput. **7**, 289–329 (2015)
23. Frasch, J.V., Vukov, M., Ferreau, H.J., Diehl, M.: A new quadratic programming strategy for efficient sparsity exploitation in SQP-based nonlinear MPC and MHE. In: Proceedings of the 19th World Congress, The International Federation of Automatic Control. Cape Town, South Africa (2014)
24. Frasch, J.V., Wirsching, L., Sager, S., Bock, H.G.: Mixed-level iteration schemes for nonlinear model predictive control. In: 4th IFAC Conference on Nonlinear Model Predictive Control. IFAC Proceedings Volumes, vol. 45, pp. 138–144. Elsevier (2012)
25. Frison, G.: Numerical Methods for Model Predictive Control. Master's thesis, Department of Informatics and Mathematical Modelling Technical University of Denmark (2012)
26. Frison, G., Jørgensen, J.B.: Efficient implementation of the Riccati recursion for solving linear-quadratic control problems. In: 2013 IEEE International Conference on Control Applications (CCA), pp. 1117–1122 (2013)
27. Frison, G., Jørgensen, J.B.: A fast condensing method for solution of linear-quadratic control problems. In: Proceedings of the IEEE Conference on Decision and Control (CDC), pp. 7715–7720. IEEE (2013)
28. Frison, G., Kouzoupis, D., Jørgensen, J.B., Diehl, M.: An efficient implementation of partial condensing for nonlinear model predictive control. In: Proceedings of the IEEE Conference on Decision and Control (CDC) IEEE (2016)
29. Frison, G., Kouzoupis, D., Sartor, T., Zanelli, A., Diehl, M.: BLASFEO: Basic linear algebra subroutines for embedded optimization. ACM Trans. Math. Softw. Accepted (2018)

30. Frison, G., Sorensen, H.B., Dammann, B., Jørgensen, J.B.: High-performance small-scale solvers for linear model predictive control. In: Proceedings of the European Control Conference (ECC), pp. 128–133 (2014)
31. Frison, G.: Algorithms and Methods for High-Performance Model Predictive Control. PhD thesis, Technical University of Denmark (DTU) (2015)
32. Gafni, E.M., Bertsekas, D.P.: Two-metric projection methods for constrained optimization. SIAM J. Control Optim. **22**, 936–964 (1984)
33. Gertz, E.M., Wright, S.J.: Object-oriented software for quadratic programming. ACM Trans Math. Softw. **29**, 58–81 (2003)
34. Gill, P.E., Murray, W., Saunders, M.A.: User's guide for QPOPT 1.0: a fortran package for quadratic programming (1995)
35. Giselsson, P.: Improved fast dual gradient methods for embedded model predictive control. In: Proceedings of the IFAC World Congress, vol. 47, pp. 2303–2309 (2014)
36. Glad, T., Johnson, H.: A method for state and control constrained linear-quadratic control problems. In: Proceedings of the IFAC World Congress, pp. 1583–1587 (1984)
37. Gondzio, J., Grothey, A.: A new unblocking technique to warmstart interior point methods based on sensitivity analysis. SIAM J. Control Optim. **19**, 1184–1210 (2006)
38. Gros, S., Zanon, M., Quirynen, R., Bemporad, A., Diehl, M.: From linear to nonlinear MPC: bridging the gap via the real-time iteration. Int. J Control. https://doi.org/10.1080/00207179.2016.1222553 (2016)
39. Hours, J.H., Jones, C.N.: A parametric nonconvex decomposition algorithm for real-time and distributed NMPC. IEEE Trans. Autom. Control **61**, 287–302 (2014)
40. Houska, B., Ferreau, H.J., Diehl, M.: ACADO toolkit—an Open source framework for automatic control and dynamic optimization. Optim. Control Appl. Methods **32**, 298–312 (2011)
41. Houska, B., Ferreau, H.J., Diehl, M.: An auto-generated real-time iteration algorithm for nonlinear MPC in the microsecond range. Automatica **47**, 2279–2285 (2011)
42. Houska, B., Frasch, J.V., Diehl, M.: An augmented Lagrangian based algorithm for distributed nonconvex optimization. SIAM J. Optim. **26**, 1101–1127 (2016)
43. Janka, D., Kirches, C., Sager, S., Wächter, A.: An SR1/BFGS SQP algorithm for nonconvex nonlinear programs with block-diagonal hessian matrix. Math. Program. Comput. **8**, 435–459 (2016)
44. Käpernick, B., Graichen, K.: The gradient based nonlinear model predicitive control software GRAMPC. In: Proceedings of the European Control Conference (ECC) (2014)
45. Kirches, C., Bock, H.G., Schlöder, J.P., Sager, S.: Complementary condensing for the direct multiple shooting method. In: Bock, H.G., Phu, H.X., Rannacher, R., Schlöder, J. (eds.) Modeling, Simulation and Optimization of Complex Processes, pp. 195–206. Springer, Berlin (2012)
46. Kirches, C., Sager, S., Bock, H.G., Schlöder, J.P.: Time-optimal control of automobile test drives with gear shifts. Optim. Control Appl. Methods **31**, 137–153 (2010)
47. Kirches, C., Wirsching, L., Sager, S., Bock, H.G.: Efficient numerics for nonlinear model predictive control. In: Diehl, M., Glineur, F., Jarlebring, E., Michiels, W. (eds.) Recent Advances in Optimization and Its Applications in Engineering, pp. 339–357. Springer, Berlin (2010)
48. Kouzoupis, D., Klintberg, E., Diehl, M., Gros, S.: A dual Newton stratgy for scenario decomposition in robust multistage MPC. Int. J. Robust Nonlinear Control **28**, 1913–2677 (2017)
49. Kouzoupis, D., Quirynen, R., Frasch, J.V., Diehl, M.: Block condensing for fast nonlinear MPC with the dual Newton strategy. In: Proceedings of the IFAC Conference on Nonlinear Model Predictive Control (NMPC), vol. 48, pp. 26–31 (2015)
50. Leineweber, D.B.: Analyse und Restrukturierung eines Verfahrens zur Direkten Lösung von Optimalsteuerungsproblemen. Master's thesis, University of Heidelberg (1995)
51. Leineweber, D.B.: Efficient reduced SQP methods for the optimization of chemical processes described by large sparse DAE models. Fortschritt-Berichte VDI Reihe 3 Verfahrenstechnik, vol. 613. VDI Verlag, Düsseldorf (1999)
52. Leineweber, D.B., Bauer, I., Bock, H.G., Schlöder, J.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part I: Theoretical aspects. Comput. Chem. Eng. **27**, 157–166 (2003)
53. Leineweber, D.B., Schäfer, A., Bock, H.G., Schlöder, F.P.: An efficient multiple shooting based reduced SQP strategy for large-scale dynamic process optimization. Part II: Software aspects and applications. Comput. Chem. Eng. **27**, 167–174 (2003)
54. Leineweber, D.B., Bock, H.G., Schlöder, J.P.: Fast direct methods for real-time optimization of chemical processes. In: Proceedings of the 15th IMACS World Congress on Scientific Computation, Modelling and Applied Mathematics Berlin, Berlin, 1997. Wissenschaft- und Technik-Verlag (1997)
55. Li, W.C., Biegler, L.T.: Multistep, Newton-type control strategies for constrained nonlinear processes. Chem. Eng. Res. Des. **67**, 562–577 (1989)

56. Mattingley, J., Boyd, S.: CVXGEN: A code generator for embedded convex optimization. Optim. Eng. **13**, 1–27 (2012)
57. Mayne, D.Q.: Model predictive control: recent developments and future promise. Automatica **50**, 2967–2986 (2014)
58. Nesterov, Y.: Introductory Lectures on Convex Optimization: A Basic Course. Applied Optimization. vol. 87. Kluwer Academic Publishers (2004)
59. Nocedal, J., Wright, S.J.: Numerical Optimization, 2nd edn. Springer Series in Operations Research and Financial Engineering. Springer-Verlag, New York (2006)
60. Ohtsuka, T.: Acontinuation/GMRES method for fast computation of nonlinear receding horizon control. Automatica **40**, 563–574 (2004)
61. Patrinos, P., Bemporad, A.: An accelerated dual gradient-projection algorithm for linear model predictive control. In: Proceedings of the IEEE Conference on Decision and Control (CDC). IEEE (2012)
62. Patrinos, P., Bemporad, A.: An accelerated dual gradient-projection algorithm for embedded linear model predictive control. IEEE Trans. Autom. Control **59**, 18–33 (2014)
63. Patrinos, P., Sopasakis, P., Sarimveis, H.: A global piecewise smooth Newton method for fast large-scale model predictive control. Automatica **47**, 2016–2022 (2011)
64. Powell, M.J.D.: A fast algorithm for nonlinearly constrained optimization calculations. In: Watson, G. (ed.) Numerical Analysis, Dundee 1977. Lecture Notes in Mathematics, vol. 630, pp. 144–157. Springer, Berlin (1978)
65. Qin, S.J., Badgwell, T.A.: A survey of industrial model predictive control technology. Control Eng. Pract. **11**, 733–764 (2003)
66. Quirynen, R., Vukov, M., Diehl, M.: Multiple shooting in a microsecond. In: Carraro, T., Geiger, M., Körkel, S., Rannacher, R. (eds.) Multiple Shooting and Time Domain Decomposition Methods, pp. 183–201. Springer, Switzerland (2015)
67. Quirynen, R., Vukov, M., Zanon, M., Diehl, M.: Autogenerating microsecond solvers for nonlinear MPC: A tutorial using ACADO integrators. Optimal Control Appl. Methods **36**, 685–704 (2014)
68. Rao, C.V., Wright, S.J., Rawlings, J.B.: Application of interior-point methods to model predictive control. J. Optim. Theory Appl. **99**, 723–757 (1998)
69. Rawlings, J.B., Mayne, D.Q., Diehl, M.M.: Model Predictive Control: Theory, Computation, and Design, 2nd edn. Nob Hill Publishing (2017)
70. Richter, S.: Computational Complexity Certification of Gradient Methods for Real-time Model Predictive Control. PhD thesis, ETH Zürich (2012)
71. Richter, S., Jones, C.N., Morari, M.: Real-time input-constrained MPC using fast gradient methods. In: Proceedings of the IEEE Conference on Decision and Control (CDC) IEEE (2009)
72. Sager, S.: Numerical Methods for Mixed–Integer Optimal Control Problems. Der Andere Verlag, Lübeck (2005)
73. Sargent, R.W.H., Sullivan, G.R.: The development of an efficient optimal control package. In: Stoer, J. (ed.) Optimization Techniques: Proceedings of the 8th IFIP Conference on Optimization Techniques Würzburg, September 5–9, 1977, pp. 158–168. Springer-Verlag (1978)
74. Schmid, C., Biegler, L.T.: Quadratic programming methods for reduced Hessian SQP. Comput. Chem. Eng. **18**, 817–832 (1994)
75. Shahzad, A., Goulart, P.J.: A new hot-start interior-point method for model predictive control. In: Proceedings of the IFAC World Congress (2011)
76. Steinbach, M.C.: A structured interior point SQP method for nonlinear optimal control problems. In: Bulirsch, R., Kraft, D. (eds.) Control, Computation Optimal, pp. 213–222. Birkhäuser, Boston (1994)
77. Stella, L., Themelis, A., Sopasakis, P., Patrinos, P.: A simple and efficient algorithm for nonlinear model predictive control. In: Proceedings of the IEEE Conference on Decision and Control (CDC), pp. 1939–1944. IEEE (2017)
78. Stellato, B., Banjac, G., Goulart, P., Bemporad, A., Boyd, S.: OSQP: an operator splitting solver for quadratic programs. arXiv:1711.08013 (2017)
79. Ullmann, F.: Fiordos: A Matlab Toolbox for C-Code Generation for First Order Methods. Master's thesis, ETH Zurich (2011)
80. Vukov, M., Domahidi, A., Ferreau, H.J., Morari, M., Diehl, M.: Auto-generated algorithms for nonlinear model predictive control on long and on short horizons. In: Proceedings of the IEEE Conference on Decision and Control (CDC), pp. 5113–5118. IEEE (2013)
81. Wang, Y., Boyd, S.: Fast model predictive control using online optimization. IEEE Trans. Control Syst. Technol. **18**, 267–278 (2010)
82. Wirsching, L.: Multi-level Iteration Schemes with Adaptive Level Choice for Nonlinear Model Predictive Control. PhD thesis, Ruprecht-Karls-Universität, Heidelberg (2018)

83. Wirsching, L., Bock, H.G., Diehl, M.: Fast NMPC of a chain of masses connected by springs. In: Proceedings of the IEEE International Conference on Control Applications, Munich, pp. 591–596. IEEE (2006)
84. Wright, S.J.: Partitioned dynamic programming for optimal control. SIAM J. Optim. **1**, 620–642 (1991)
85. Wright, S.J.: Structured interior point methods for optimal control. In: Proceedings of the 30th IEEE Conference on Decision and Control, pp. 1711–1716 (1991)
86. Wright, S.J.: Primal-Dual Interior-Point Methods. SIAM Publications, Philadelphia (1997)
87. Zanelli, A., Domahidi, A., Jerez, J.L., Morari, M.: FORCES NLP: An Efficient implementation of interior-point methods for multistage nonlinear nonconvex programs. Int. J Control. https://doi.org/10.1080/00207179.2017.1316017 (2017)
88. Zanelli, A., Quirynen, R., Jerez, J., Diehl, M.: A homotopy-based nonlinear interior-point method for NMPC. In: Proceedings of the IFAC World Congress, Toulouse, France (2017)
89. Zavala, V.M., Biegler, L.T.: The advanced step NMPC controller: Optimality, stability and robustness. Automatica **45**, 86–93 (2009)
90. Zometa, P., Kögel, M., Findeisen, R.: μAO-MPC: A free code generation tool for embedded real-time linear model predictive control. In: 2013 American Control Conference, pp. 5320–5325. IEEE, Washington, DC (2013)
91. ACADOS. https://github.com/acados/acados (2018)
92. BLASFEO. https://github.com/giaf/blasfeo (2016)
93. hangingchainbenchmark. https://github.com/dkouzoup/hanging-chain-acado (2018)
94. HPIPM. https://github.com/giaf/hpipm (2017)
95. HPMPC. https://github.com/giaf/hpmpc (2014)
96. treeQP. https://github.com/dkouzoup/treeQP (2017)